

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

KI-SEON YANG

Serial No.: *to be assigned*

Examiner: *to be assigned*

Filed: 16 March 2004

Art Unit: *to be assigned*

For: CONFIGURING DIRECTION-BASED CORE BASED TREE (CBT) FOR
CBT-BASED OVERLAY MULTICAST

**CLAIM OF PRIORITY
UNDER 35 U.S.C. §119**

Mail Stop Patent Application

Commissioner for Patents

P.O.Box 1450

Alexandria, VA 22313-1450

Sir:

The benefit of the filing date of the following prior foreign application, Korean Priority No. 2003-19718 (filed in Korea on 28 March 2003, and filed in the U.S. Patent and Trademark Office on 16 March 2004), is hereby requested and the right of priority provided in 35 U.S.C. §119 is hereby claimed.

In support of this claim, filed herewith is a certified copy of said original foreign application.

Respectfully submitted,



Robert E. Bushnell
Reg. No.: 27,774
Attorney for the Applicant

Suite 300, 1522 "K" Street, N.W.
Washington, D.C. 20005
(202) 408-9040

Folio: P57003
Date: 16 March 2004
I.D.: REB/kf



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.

출원번호 : 10-2003-0019718
Application Number

출원년월일 : 2003년 03월 28일
Date of Application MAR 28, 2003


출원인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003 년 07 월 08 일

특허청

COMMISSIONER



【서지사항】

【서류명】	특허출원서		
【권리구분】	특허		
【수신처】	특허청장		
【제출일자】	2003.03.28		
【발명의 명칭】	씨비티 기반 오버레이 멀티 캐스트를 위한 방향성 기반 씨비티 구성방법		
【발명의 영문명칭】	method for construction of CBT direction based for overlay multicast CBT based		
【출원인】			
【명칭】	삼성전자 주식회사		
【출원인코드】	1-1998-104271-3		
【대리인】			
【성명】	박상수		
【대리인코드】	9-1998-000642-5		
【포괄위임등록번호】	2000-054081-9		
【발명자】			
【성명의 국문표기】	양기선		
【성명의 영문표기】	YANG,KI SEON		
【주민등록번호】	690510-1105820		
【우편번호】	442-725		
【주소】	경기도 수원시 팔달구 영통동 벽적골8단지아파트 한신아파트 813동 1 806호		
【국적】	KR		
【심사청구】	청구		
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사 를 청구합니다. 대리인 박상수 (인)		
【수수료】			
【기본출원료】	20	면	29,000 원
【가산출원료】	29	면	29,000 원
【우선권주장료】	0	건	0 원
【심사청구료】	5	항	269,000 원
【합계】	327,000 원		
【첨부서류】	1. 요약서·명세서(도면)_1통		

【요약서】

【요약】

본 발명에 따른 CBT 기반 오버레이 멀티 캐스트를 위한 방향성 기반 CBT 구성방법은 CBT에 가입하고자 하는 임의의 종단 노드가 코어 노드에 기가입된 자식 노드 정보를 요청하여 수신하는 단계와, 코어 노드로부터 임의의 자식 노드 정보가 수신한 종단 노드가 자신과 수신된 각 자식 노드와의 방향성을 계산하여 최소의 결과값을 갖는 코어 노드의 자식 노드에 대한 정보를 가입 요구 메시지와 함께 코어 노드에 전송하는 단계와, 코어 노드가 종단 노드로부터 전송된 가입 요구 메시지 및 자식 노드에 대한 정보를 수신하여 해당 자식 노드와 종단 노드의 방향성 계산 결과값을 기가입된 자식 노드간의 방향성 계산 결과값과 비교하여 그 비교 결과에 따라 종단 노드를 해당 자식 노드의 부모 노드 또는 자식 노드로 가입시켜 CBT를 구성하는 단계와, 코어 노드 및 종단 노드가 부모 노드, 자식 노드, 형제 노드와 주기적으로 헬로우 패킷을 송수신하여 해당 노드의 상태를 확인하여 그 상태값에 따라 구성된 CBT를 재구성하는 단계를 수행함으로써, 특정 라우터에서의 혼잡 발생 가능성을 감소시킴으로써 보다 안정적인 멀티캐스트를 제공할 수 있다.

【대표도】

도 8

【색인어】

CBT, 멀티 캐스트, 오버레이 멀티 캐스트, 노드, 방향성, 거리

【명세서】

【발명의 명칭】

씨비티 기반 오버레이 멀티 캐스트를 위한 방향성 기반 씨비티 구성방법{method for construction of CBT direction based for overlay multicast CBT based}

【도면의 간단한 설명】

도 1a는 멀티 캐스트 라우터를 이용한 IP 멀티 캐스트를 수행하는 네트워크의 형상도.

도 1b는 유니 캐스트 라우터를 이용한 오버레이 멀티 캐스트를 수행하는 네트워크의 형상도.

도 2는 동일한 멀티 캐스트 그룹에 속한 노드들로 이루어지는 네트워크 토플로지.

도 3a 내지 도 3d는 도 2의 네트워크 토플로지에서 코어 노드가 C0인 거리 메트릭 기반 CBT에 의거하여 노드 a, b, h의 순서로 가입하는 경우를 설명하기 위한 도면.

도 4는 도 3c의 CBT에서의 멀티 캐스트 패킷의 중계경로도.

도 5a는 C0, a, b, h, h1의 노드로 구성된 거리 메트릭 기반의 CBT 구성도.

도 5b는 도 5a의 네트워크에서 각 링크 및 라우터에서 처리하는 패킷의 수를 나타내는 그래프.

도 6a 내지 도 6d는 코어 노드를 기준으로 이웃하는 두 노드간의 위치관계를 보여주는 도면.

도 7a 내지 도 7c는 본 발명에 따른 코어 노드가 C0인 방향성 메트릭 기반 CBT에 의거하여 노드 a, b, h의 순서로 가입하는 경우를 설명하기 위한 도면.

도 8은 본 발명에 따른 방향성 메트릭 기반 CBT에서의 멀티 캐스트 패킷의 중계 경로도.

도 9a 내지 도 9b는 본 발명에 따른 코어 노드가 C0인 방향성 메트릭 기반 CBT에 의거하여 노드 a, b, h, h1의 순서로 가입하는 경우를 설명하기 위한 도면.

도 10은 링 토플로지 네트워크 구성도.

도 11a 및 도 11b는 표 2에서 실험1에 대한 거리 메트릭 기반 CBT와 본 발명에 따른 방향성 메트릭 기반 CBT 구성도.

도 12는 도 11a 및 도 11b에서 종단 노드들과 연결된 각 라우터에서 처리하는 패킷 수를 비교한 그래프.

도 13a 및 도 13b는 실험 1에서 각 노드들이 CBT에 가입하는 경우를 실험한 결과의 평균처리 패킷수를 나타내는 그래프.

도 14는 선형 토플로지 네트워크 구성도.

도 15a 및 도 15b는 실험 2에서 각 노드별 CBT 가입 실험 결과의 평균처리 패킷수를 나타내는 그래프.

도 16은 성형 토플로지 네트워크 구성도.

도 17a 및 도 17b는 실험 3에서 각 노드들이 CBT에 가입하는 경우를 실험한 결과의 평균처리 패킷수를 나타내는 그래프.

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<20> 본 발명은 CBT(Core Based Tree) 기반 오버레이 멀티 캐스트를 위한 방향성 기반 CBT 구성방법에 관한 것이다.

<21> 동일한 정보를 여러 수신자에게 송신하고자 할 경우는 일대일 통신인 유니캐스트 (unicast)보다 멀티 캐스트(multicast)를 통해 해당 멀티캐스트 그룹에 참가한 수신자들에게만 데이터를 전송하는 일대 다 통신이 송신자의 전송 오버헤드, 네트워크의 자원, 수신자 측에서의 지연(latency)를 줄일 수 있어 효율적이다.

<22> 그러나, 네트워크에서 수행되는 IP 멀티캐스트는 멀티 캐스트 라우팅 알고리즘에 따라 데이터를 중계하는 멀티 캐스트 라우터로 네트워크를 구성해야 한다.

<23> 이에 따라, 기존의 유니 캐스트 라우터를 멀티 캐스트 라우터로 교체해야 하는 신규 투자 비용이 소요되는 문제가 발생한다.

<24> 한편, ISP 입장에서는 멀티캐스트 서비스 수익 모델 부재로 인한 투자대비 수익성의 이유로, 또한 기술적인 측면에서도 SIP간 멀티 캐스트 트래픽 통과에 대한 협상과 정책, 기존 유니 캐스트 트래픽과의 Fairness, 그리고 여전히 불투명한 Inter-Domain 멀티 캐스트 기술등으로 특히 ISP 백본망에서 멀티 캐스트 도입이 지연되고 있다.

<25> 반면에, 인터넷을 이용한 사내방송, 캠퍼스 방송등 사적이고 지역적인 멀티 캐스트 요구는 급속히 증가하고 있다. 이에 따라 현재의 유니캐스트 망을 기반으로 하는 멀티캐스트 전송의 필요성이 대두되고 있다.

<26> 유니 캐스트망 기반의 멀티 캐스트 전송의 한 방법으로, 종단 노드에서 데이터를 중계(relay)하는 오버레이(overlay) 멀티캐스트가 있다. 종단 노드는 최근 데이터 처리 속도의 급격한 향상, 메모리의 증가 및 VDSL 보급 등으로 이루어진 데이터 전송속도의 급속한 향상으로 다른 종단 노드로 데이터를 중계하는 것이 가능하여 효율적인 오버레이 멀티캐스트를 기대할 수 되었다.

<27> 도 1a는 멀티 캐스트 라우터를 이용한 IP 멀티 캐스트를 수행하는 네트워크를 나타내고, 도 1b는 유니 캐스트 라우터를 이용한 오버레이 멀티 캐스트를 수행하는 네트워크를 나타낸다.

<28> 오버레이 멀티 캐스트의 기본 개념은 도 1a에 도시된 바와 같이 멀티 캐스트 라우터들(Multicast Router)(MR1 - MR4)로 구성된 네트워크에서 멀티 캐스트를 수행하는 대신에 도 1b에 도시된 바와 같이 유니 캐스트 라우터들(Unicast Router)(R1 - R4)로 구성된 네트워크에서 노드 B와 같이 종단 노드(호스트)에서 패킷을 중계(복사한 후 전송)하는 방법으로 유니 캐스트 기반의 멀티 캐스트를 수행하는 것이다. 즉, 종단 노드에서 멀티 캐스트 라우터의 역할을 수행하는 것이다.

<29> 도 1b의 네트워크에서 오버레이 멀티 캐스트를 수행하는 것이 도 1a의 네트워크에서 멀티 캐스트를 수행하는 것보다 지연시간이나 패킷수가 증가하지만, 멀티 캐스트 라우터를 필요로 하지 않아 기존의 유니 캐스트망 기반에서 비교적 효율적인 멀티 캐스트 통신을 제공할 수 있다.

<30> 오버레이 멀티 캐스트 프로토콜 관련 연구로는 DVMRP(Distance Vector Multicast Routing Protocol), 멀티 캐스트 라우팅 프로토콜을 적용한 Narada, 네트워크상에

scattercast proxy를 설치하여 proxy간의 유니 캐스트 통신을 이용한 scattercast, 그리고, CBT(Core Based Tree) 기반의 연구들이 진행중에 있다.

<31> 이러한 오버레이 멀티 캐스트의 핵심은 동일한 멀티 캐스트 그룹에 속한 종단 노드들로 효율적인 패킷 전송을 위한 오버레이 트리를 구성하고 유지하는 것이다.

<32> 한편, DVMRP 멀티 캐스트 라우팅 프로토콜을 적용한 경우보다 공유트리인 CBT를 적용하는 경우가 확장성과 제어 메시지의 오버헤드(OVERHEAD), 그리고 성능면에서 효율적이다.

<33> 즉, CBT는 코어 노드로 트래픽이 집중되는 단점이 있으나, 멀티 캐스트 멤버의 변동에 대하여 상대적으로 적은 제어 메시지를 발생하고 각 종단 노드에서 관리해야 할 정보량이 적어 확장성이 뛰어난 장점이 있고 구현이 쉽다. 또한 종단 노드의 처리속도 향상과 데이터 전송 능력 향상에 비례하여 중계 가능한 자식 수를 증가시킴으로써 전체 멀티 캐스트의 성능을 향상시킬 수 있는 장점으로 인하여 CBT로 오버레이 트리를 구성하는 것이 보다 효율적이다.

<34> CBT 기반 오버레이 멀티캐스트는 기본적으로 노드간의 거리(distance)가 가까운 노드를 부모 또는 자식 노드로 선택하여 CBT를 구성한다. 거리를 메트릭(Metric)으로 구성하는 CBT에서는 CBT상의 전체 노드들로 패킷을 중계하는 과정에서 특정 링크나 라우터를 중복하여 경유함으로써, 해당 링크나 라우터에 트래픽이 집중되어 혼잡(Congestion)이 발생할 수 있다.

<35> 일반적으로 멀티 캐스트는 TCP와 달리 혼잡 제어 메카니즘을 갖고 있지 않는 UDP를 기반으로 동작하므로 네트워크에 혼잡이 발생하는 경우에도 트래픽 양을 감소하지 않고

계속해서 패킷을 네트워크로 전송한다. 그 결과 패킷의 손실로 인한 멀티캐스트의 성능 뿐만 아니라, 네트워크의 성능을 악화시키고 네트워크상의 다른 TCP 연결들과의 공평성 문제(Unfair problem)를 일으킬 수 있는 문제점이 있다.

<36> 도 2는 동일한 멀티 캐스트 그룹에 속한 노드들로 이루어지는 네트워크 토플로지의

일예로서 각 노드의 가입순서는 a, b, h, h1의 순서로 이루어진다.

<37> 도 2를 참조하면, 코어 노드 C0에 링크 #1을 통해 라우터1(R1)이 연결되고, 노드 a는 라우터1(R1)에 링크 #2를 통해 연결된다. 라우터1(R1)에 링크 #3을 통해 라우터3(R3)이 연결되고, 노드 b가 링크 #32를 통해 라우터3(R3)에 연결된다. 노드 h는 링크 #31를 통해 라우터3(R3)에 연결되고, 노드 h1은 링크 #33을 통해 라우터3(R3)에 연결되어 있다

<38> 도 3a 내지 도 3d는 도 2의 네트워크 토플로지에서 코어 노드가 C0인 거리 메트릭 기반 CBT에 의거하여 노드 a, b, h의 순서로 가입하는 경우를 설명하기 위한 도면이다.

<39> 도 3a은 도 2의 네트워크에서 코어 노드가 C0인 거리 메트릭 기반 CBT에 노드 a, b, h의 순서로 가입하는 경우에 각 노드간의 거리를 나타낸다. 코어 노드 C0과 노드 a의 거리는 3이고, 코어 노드 C0과 노드 b의 거리는 9이고, 노드 a와 노드 h의 거리는 4이고, 노드 h와 노드 b와의 거리는 8이고, 노드 a와 노드 b의 거리는 10이다.

<40> 즉, 코어 노드 C0과 노드 a는, 링크 #1를 통해 라우터 1(R1)을 경유하여 링크 #2를 통해 연결됨으로 링크 #1의 거리 1과, 링크 #2의 거리 2를 더하면 전체적인 거리는 3이 된다.

<41> 코어 노드 C0과 노드 b는 링크 #1을 통해 라우터 1(R1)를 경유하고, 링크 #3을 통해 라우터 3(R3)을 경유하고, 링크 #32를 통해 연결됨으로, 링크 #1의 거리 1과, 링크 #3의 거리 1과, 링크 32의 거리 7를 더하면 전체적인 거리는 9가 된다.

<42> 노드 a와 노드 h는 링크 #2를 통해 라우터1(R1)을 경유하고, 링크 #3을 경유하여 라우터3(R3)를 경유하고, 링크 #31를 통해 연결됨으로, 링크 #2의 거리 2와, 링크 #3의 거리 1과, 링크 #31의 거리 1을 더하면 전체적인 거리는 4가 된다.

<43> 노드 h와 노드 b는 링크 #32를 통해 라우터3(R3)을 경유하여 링크 #31를 통해 연결되므로, 링크 #32의 거리 7과 링크 #31의 거리 1을 더하면 전체적인 거리는 8이 된다.

<44> 노드 a와 노드 b는 링크 #2를 통해 라우터1(R1)을 경유하고, 링크 #3을 통해 라우터 3(R3)을 경유하고, 링크 #32를 통해 연결되므로, 링크 #2의 거리 2와, 링크 #3의 거리 1과, 링크 #32의 거리 7을 더하면 전체적인 거리는 10이 된다.

<45> 여기에서 두 노드간 거리는 질의(Query)와 응답(response)의 교환에 대한 종단간 지연시간(end-to-end-latency)을 이용하였다.

<46> 코어 C0의 수용(중계) 가능한 자식 수가 2인 경우, 도 2와 도 3a를 참조하면, 도 3b에 도시된 바와 같이 먼저 a와 b 노드가 가입된 상태에서, h노드는 a노드와는 거리가 4이고 b노드와는 8이므로 a노드를 부모노드로 선택해 가입하여 도 3c의 CBT를 구성한다.

<47> 도 3d는 코어 노드로부터 하나의 멀티 캐스트 패킷이 전체 CBT로 중계되는 과정에서 각 링크 및 라우터에서 처리하는 패킷의 수를 보여준다.

<48> 즉, 도 2의 네트워크 토플로지에서 도 3a와 같은 거리를 가짐에 따라, 도 3b 및 도 3c와 같은 절차를 거치는 거리 기반 CBT를 구성하면 각 링크 및 라우터에서 처리하는 패킷수는 다음과 같다.

<49> 코어 노드 C0에서 각 노드 a, 노드 b, 노드 h로 패킷을 전달하기 위해서는 코어 노드에서 노드 a와 노드 b로 패킷을 전송하고, 노드 a노드에서 패킷을 중계하여 노드 h로 보내줄 수 있다.

<50> 즉, 도 2를 참조하여 코어 노드 C0에서 노드 a까지의 패킷전송 경로를 살펴보면, 코어노드 C0에서 링크 #1을 통해 라우터1(R1)를 경유하여 링크 #2를 통해 노드 a까지의 경로가 그려진다.(링크 #1, R1, 링크 #2)

<51> 코어 노드 C0에서 노드 b까지의 패킷 전송 경로를 살펴보면, 코어 노드 C0에서 링크 #1을 통해 라우터1(R1)을 경유하고, 링크 #3을 통해 라우터3(R3)를 경유하고, 링크 #32를 통해 노드 b까지의 경로가 그려진다.(링크 #1, R1, 링크 3, R3, 링크 #32)

<52> 노드 a에서 패킷을 중계하여 노드 h로 보내주는 경로를 살펴보면, 노드 a에서 링크 #2를 통하여 라우터 1(R1)를 경유하고, 링크 #3을 통해 라우터 3(R3)을 경유하고, 링크 #31을 통해 노드 h까지의 경로가 그려진다.(링크 #2, R1, 링크 #3, R3, 링크 #31)

<53> 따라서, 각 링크 별 라우터별로 패킷이 통과하는 횟수를 계산해보면, 링크 #1은 2회, 라우터1(R1)은 3회, 링크 #2는 2회, 링크 #3은 2회, 라우터3(R3)은 2회, 링크 #31은 1회, 링크 #32는 1회가 된다.

<54> 즉, N개의 멀티 캐스트 패킷을 전체 CBT로 전송하기 위하여 라우터1(R1)에서 3*N개의 패킷을 처리한다.

<55> 이처럼 다른 링크나 라우터보다 R1에 패킷이 집중되는 것은 도 4에서 보는 바와 같이 패킷을 첫번째 계층인 C0에서 R1을 경유하여 노드 a와 노드 b로 중계하는 경로(실선 화살표)에 대하여 두번째 계층 노드 a에서 노드 h로 패킷을 중계하는 경로(점선 화살표)가 R1을 경유하여 되돌아가기 때문이다.

<56> 즉, 실선 화살표로 도시된 바와 같이 C0에서 노드 a와 노드 b로 중계할 때 각각 한번 씩 R1을 경유하게 되고, 점선 화살표로 도시된 바와 같이 노드 a에서 노드 b로 패킷을 중계할 때도 R1 및 R3를 경유하게 됨으로 R1을 전체적으로 3번을 경유하게 된다.

<57> 도 5a는 도 2에 도시된 네트워크 토플로지에서 C0, a, b, h, h1의 노드로 구성된 거리 메트릭 기반의 CBT를 나타내는 것으로 노드 h1은 노드 a를 부모노드로 선택해 가입한다. 도 5b는 도 5a의 네트워크에서 각 링크 및 라우터에서 처리하는 패킷의 수를 나타낸다. 도 5b를 참조하면, 각 링크 #1은 2개, 라우터 1(R1)은 4개, 링크 #2는 3개, 링크 #3은 3개, 라우터 3(R3)은 3개, 링크 #31은 1개, 링크 #32는 1개, 링크 #33은 1개의 패킷을 처리한다.

<58> 즉, 도 2의 네트워크 토플로지에서 도 3a와 같은 거리를 가짐에 따라, 도 5a에 도시된 바와 같은 거리 기반 CBT를 구성하면 각 링크 및 라우터에서 처리하는 패킷수는 다음과 같다.

<59> 코어 노드 C0에서 각 노드 a, 노드 b, 노드 h, 노드 h1으로 패킷을 전달하기 위해서는 코어 노드에서 노드 a와 노드 b로 패킷을 전송하고, 노드 a노드에서 패킷을 중계하여 노드 h와 노드 h1으로 보내줄 수 있다.

<60> 즉, 도 2를 참조하여 코어 노드 C0에서 노드 a까지의 패킷전송 경로를 살펴보면, 코어노드 C0에서 링크 #1을 통해 라우터1(R1)를 경유하여 링크 #2를 통해 노드 a까지의 경로가 그려진다.(링크 #1, R1, 링크 #2)

<61> 코어 노드 C0에서 노드 b까지의 패킷 전송 경로를 살펴보면, 코어 노드 C0에서 링크 #1을 통해 라우터1(R1)을 경유하고, 링크 #3을 통해 라우터3(R3)를 경유하고, 링크 #32를 통해 노드 b까지의 경로가 그려진다.(링크 #1, R1, 링크 3, R3, 링크 #32)

<62> 노드 a에서 패킷을 중계하여 노드 h로 보내주는 경로를 살펴보면, 노드 a에서 링크 #2를 통하여 라우터 1(R1)를 경유하고, 링크 #3을 통해 라우터 3(R3)을 경유하고, 링크 #31을 통해 노드 h까지의 경로가 그려진다.(링크 #2, R1, 링크 #3, R3, 링크 #31)

<63> 노드 a에서 패킷을 중계하여 노드 h1으로 보내주는 경로를 살펴보면, 노드 a에서 링크 #2를 통하여 라우터 1(R1)를 경유하고, 링크 #3을 통해 라우터 3(R3)을 경유하고, 링크 #33을 통해 노드 h1까지의 경로가 그려진다.(링크 #2, R1, 링크 #3, R3, 링크 #33)

<64> 따라서, 각 링크 별 라우터별로 패킷이 통과하는 횟수를 계산해보면, 링크 #1은 2회, 라우터1(R1)은 4회, 링크 #2는 3회, 링크 #3은 3회, 라우터3(R3)은 3회, 링크 #31은 1회, 링크 #32는 1회, 링크 #33는 1회가 된다.

<65> 도 5a 및 도 5b에서 볼 수 있는 바와 같이 새로운 노드의 가입으로 R1을 경유하여 되돌아가는 횟수가 증가하면 라우터1(R1)에서는 혼잡의 문제가 발생할 수 있다.

<66> 이와 같이 거리 메트릭 기반의 CBT에서는 가입순서에 따라 혼잡이 발생하는 문제점이 있다. 특히 라우터 R1과 같이 코어 노드와 연결된 라우터나 링크에서 혼잡이 발생하는 경우, 전체 CBT에 영향을 미치는 심각한 문제를 야기시킬 수 있다.

【발명이 이루고자 하는 기술적 과제】

<67> 이와 같이 종래의 거리 메트릭 기반 CBT의 문제점은 트리에서 하위(자식)노드들로 전송되는 멀티캐스트 패킷들의 중계 경로를 고려하지 않고 단순히 노드간의 거리만으로 부모 또는 자식 노드를 선택하여 CBT를 구성한 것에 기인한다.

<68> 본 발명은 이러한 문제점을 해결하기 위하여 안출된 것으로, 특정 라우터와 링크에 트래픽이 집중되는 문제를 해결하기 위한 CBT 기반 오버레이 멀티 캐스트를 위한 방향성 기반 CBT 구성방법을 제공하는데 그 목적이 있다.

【발명의 구성 및 작용】

<69> 이러한 목적을 달성하기 위한 본 발명의 일측면에 따르면, CBT 기반 오버레이 멀티 캐스트를 위한 CBT 구성방법에 있어서, CBT에 가입하고자 하는 임의의 종단 노드가 코어 노드에 기가입된 자식 노드 정보를 요청하여 수신하는 단계와, 코어 노드로부터 임의의 자식 노드 정보가 수신한 종단 노드가 자신과 수신된 각 자식 노드와의 방향성을 계산하여 최소의 결과값을 갖는 코어 노드의 자식 노드에 대한 정보를 가입 요구 메시지와 함께 코어 노드에 전송하는 단계와, 코어 노드가 종단 노드로부터 전송된 가입 요구 메시지 및 자식 노드에 대한 정보를 수신하여 해당 자식 노드와 종단 노드의 방향성 계산 결

과값을 기가입된 자식 노드간의 방향성 계산 결과값과 비교하여 그 비교 결과에 따라 종단 노드를 해당 자식 노드의 부모 노드 또는 자식 노드로 가입시켜 CBT를 구성하는 단계와, 코어 노드 및 종단 노드가 부모노드, 자식 노드, 형제 노드와 주기적으로 헬로우 패킷을 송수신하여 해당 노드의 상태를 확인하여 그 상태값에 따라 구성된 CBT를 재구성하는 단계를 수행하는 CBT 기반 오버레이 멀티 캐스트를 위한 방향성 기반 CBT 구성방법을 제공한다.

<70> 이와 같이, 본 발명에서는 코어 노드에서 각 노드로 전송되는 패킷들의 전송 진행 방향을 분석하여 진행 방향이 같은 노드를 CBT에서의 부모 또는 자식 노드로 선택함으로써, 거리 메트릭 기반 CBT의 되돌아가는 문제점을 해결할 수 있다.

<71> 우선 본 발명에 의한 방향성 메트릭 기반 CBT의 어트리뷰트의 정의하도록 한다.

<72> 표 1은 본 발명에 따른 방향성 메트릭 기반 CBT의 정의된 어트리뷰트이다.

<73> 【표 1】

notation	설명
$C(x)$	노드 x 의 자식 집합
$num(C(x))$	노드 x 의 자식 수
$max(C(x))$	노드 x 가 최대 수용(증제) 가능한 자식 수
$P_i(x)$	노드 x 의 i 번째 조상 노드 ($i > 0$)
$dist(x, y)$	노드 x 와 y 간의 거리
C_0	코어 노드
h	CBT에 가입을 원하는 새로운 노드
N_x, N_y	CBT에 이미 가입되어 있는 노드
$p(x, y)$	노드 x 와 y 중 코어 노드와의 거리가 가까운 노드 $p(x, y) \leftarrow x, \text{ if } dist(C_0, x) < dist(C_0, y)$
$c(x, y)$	노드 x 와 y 중 코어 노드와의 거리가 먼 노드 $p(x, y) \leftarrow y, \text{ if } dist(C_0, x) < dist(C_0, y)$
$atan()$	아크탄젠트값을 라디안 값으로 계산
$sqrt()$	제곱근 값 반환
π	파이 (π) 값 (≈ 3.1415926)
$WhoisChild(x)$	노드 x 에게 x 의 자식 정보를 요청
$direction(x, y)$	노드 x 와 y 간의 방향성 값을 계산

<74> 표 1을 참조하면, $C(x)$ 는 노드 x 의 자식(child) 집합이고, $num(C(x))$ 는 노드 x 의 자식수이고, $max(C(x))$ 는 노드 x 가 최대 수용(중계) 가능한 자식수이고, $P_i(x)$ 는 노드 x 의 i 번째 조상노드($i > 0$)이다. $dist(x, y)$ 는 노드 x 와 y 간의 거리를 나타내며, C_0 은 코어 노드를 나타내며, h 는 CBT에 가입을 원하는 새로운 노드를 나타내며, N_x, N_y 는 CBT에 이미 가입되어 있는 노드를 나타낸다.

<75> $p(x, y)$ 는 노드 x 와 y 중 코어 노드와의 거리가 가까운 노드를 나타내는 것으로, 예를 들어, 코어 노드와 노드 x 와의 거리($dist(C_0, x)$)가 코어 노드와 노드 y 와의 거리($dist(C_0, y)$)보다 작으면 $p(x, y)$ 는 x 가 된다.

<76> 한편, $c(x, y)$ 는 노드 x 와 y 중 코어 노드와의 거리가 먼 노드를 나타내는 것으로, 예를 들어 코어 노드와 노드 x 와의 거리($dist(C_0, x)$)가 코어 노드와 노드 y 와의 거리($dist(C_0, y)$)보다 작으면 $c(x, y)$ 는 y 가 된다. $atan()$ 은 아크 탄젠트값을 라디안 값으로 계산하는 것을 나타내고, $sqr()$ 는 제곱근 반환을 나타내며, PI 는 파이(π)값(≈ 3.1415926)을 나타내고, $WhoisChild(x)$ 는 노드 x 에게 x 의 자식 정보를 요청하는 것을 나타내고, $direction(x, y)$ 는 노드 x 와 y 간의 방향성 값을 계산하는 것을 나타낸다.

<77> 도 6a 내지 도 6d는 본 발명에 따른 방향성 기반 CBT 구성방법을 설명하기 위한 도면으로, 코어 노드를 기준으로 이웃하는 두 노드간의 위치관계를 보여준다. 도 6a는 코어 C_0 , 노드 a_1 , 노드 a_2 의 차례로 연결되어 있고 도 6b는 노드 a_1 , 코어 C_0 , 노드 a_2 의 차례로 연결되어 있다. 도 6c 및 도 6d와 같이 코어 노드와 이웃하는 두 노드들간의 거리를 변으로 하는 삼각형을 생각할 수 있다. 이 삼각형의 내외각과 높이를 이용하여 노드간 중계되는 패킷들의 진행방향성을 구한다.

<78> 본 발명에 의한 방향성 계산 알고리즘은 다음과 같이 나타낼 수 있다.

```
<79>      Procedure : direction(a1, a2)
<80>      hc ←dist( C0, c(a1, a2))
<81>      ac ←dist( C0, p(a1, a2))
<82>      ha ←dist( a1,a2)

<83>      if (hc + ac ≥ha)
<84>          retun ∞
<85>      else if (ac + ha ≤hc)
<86>          retun 0
<87>      end if
<88>      x ←((hc*hc)+(ac*ac)-(ha*ha)/(2*hc)
<89>      y ←sqrt(ac*ac - x*x)    //높이 계산
<90>      if (x > 0)
<91>          a ←atan(y/x)*(180/PI)
<92>      else
<93>          a ← 180 - atan(y/-x)*(180/PI)// 라디안 값을 각도로 변환
<94>      end if
<95>      γ ←atan(y/(hc-x))*(180/PI)
<96>      β ← a + γ
<97>      return( a + β )*y    //direction(a1, a2)의 결과 값
```

<98> 즉, 노드 a1과 노드 a2중 코어 C0에서 먼 노드와 C0의 거리를 hc라 하고, 노드 a1과 노드 a2중 코어 C0에서 가까운 노드와 C0의 거리를 ac라 하고, 노드 a1과 노드 a2의 거리를 ha라 하자.

<99> 우선 도 6a처럼 $ac + ha \leq hc$ 인 경우에는 리턴값으로 0을 출력하고, 도 6b처럼 $hc + ac \geq ha$ 인 경우에는 리턴값으로 ∞ 를 출력한다.

<100> 한편, 도 6c 또는 도 6d의 경우에는 $((hc*hc)+(ac*ac)-(ha*ha)/(2*hc))$ 를 x로 하고, $\sqrt{ac*ac - x*x}$ 를 y로 하여 높이(y)를 계산한다.

<101> 이때, 도 6c처럼 $x > 0$ 인 경우에는 $\text{atan}(y/x)*(180/\text{PI})$ 을 a로 하고, 도 6d처럼 $x < 0$ 인 경우에는 $180 - \text{atan}(y/-x)*(180/\text{PI})$ 를 a로 하여 라디안 값을 각도로 변환한다.

<102> 그 다음, $\text{atan}(y/(hc-x))*(180/\text{PI})$ 를 y로 하고, a + y를 β 로 하여 $(a + \beta)*y$ 값을 direction(a1, a2)의 결과 값으로 리턴한다.

<103> 위에서 볼 수 있는 바와 같이 본 발명에서는 방향성 계산 알고리즘으로 계산된 방향성 값을 거리 대신 새로운 메트릭으로 이용하여 CBT를 구성한다. 즉, 두 노드간의 방향성 값이 작을수록 두 노드는 코어 노드로부터 수신하는 패킷의 진행하는 방향이 동일하다고 간주하여 부모 또는 자식 노드로 선택한다. 이를 통하여 라우터 1에서 되돌아가는 횟수를 줄일 수 있다.

<104> 코어 노드 선출에 관한 방법은 수동적인 방법을 사용한다. Bootstrap Mechanism 등의 방법도 있으나 구현의 편리를 위하여 코어 노드를 지정하고, 그룹에 참가하는 모든

노드들은 설정을 통하여 코어 노드를 알고 있거나, 코어 노드를 알고 있는 노드에 대한 정보를 가진다고 가정하였다.

<105> 도 7a 내지 도 7c와 도 8 및 도 9a 및 도 9b는 본 발명에 따른 방향성 기반 CBT를 설명하기 위한 도면으로, 종래의 거리 기반 CBT 구성과 비교하기 위하여 도 2에 도시된 네트워크 토플로지에서 a, b, h, h1 노드의 순서로 CBT에 가입하는 경우를 살펴보도록 한다.

<106> 따라서, 각 도면에 도시된 각 노드간의 거리는 도 2에 도시된 네트워크 토플로지와 도 3a에 도시된 각 노드간의 거리를 나타낸 것이다. 각 노드간의 거리는 도 3a에서 설명한 바 있다.

<107> 도 7a는 h 가입전의 CBT를 나타내고, 도 7b는 h 가입후의 CBT를 나타내고, 도 7c는 8은 하나의 멀티 캐스트 패킷에 대해 각 링크 및 라우터에서 처리하는 패킷의 수를 나타낸다.

<108> 가입을 원하는 노드 h는 중계 가능한 최대 자식 수가 2인 코어 노드 C0에게 C0의 자식들인 a, b 노드 정보를 얻은 다음 a, b 노드들과 앞서 설명한 방향성 값 direction(h, a)[≈ 214.09]와 direction(h, b)[≈ 41.37]을 계산한다.

<109> h 노드는 작은 방향성 값을 나타내는 b노드와의 정보를 담은 CBT 가입 요구 메시지를 전송하여 첫번째 계층의 부모인 코어 노드 C0에게 가입을 요구한다. 코어 노드 C0은 수신한 CBT 가입 요구 메시지의 direction(h, b)값과 주기적인 헬로우(hello) 메시지를 통하여 수신한 기존의 자식 노드간 direction(a, b)[≈ 71.57]값을 비교한다. 이 결과

[direction(h, b) < direction(a, b)]는 h노드에서 b 노드로 패킷을 중계하는 경로가 앞서 설명한 되돌아가는 현상을 피하는 효율적인 경로임을 의미한다.

<110> 이에 따라 코어 노드 C0은 h 노드를 새로운 자식 노드로 가입시키고, 기존의 자식 이던 b 노드에게 새로운 부모 h 노드에 재가입할 것을 플러쉬(Flush) 메시지를 전송하여 요청한다. 이 요청에 따라 b노드는 두번째 계층의 부모 h노드에 가입하여 도 7b와 같은 CBT를 구성한다.

<111> 도 7c는 각 링크 및 라우터가 처리하는 패킷의 개수이다. 도면에서 실선은 본 발명에 따른 방향성 기반 CBT 구성에 대한 결과이고, 점선은 종래의 거리 기반 CBT 구성에 대한 결과이다. 도 7c를 참조하여 본 발명의 결과를 살펴보면, 도면에서 실선으로 도시된 바와 같이 각각 링크 #1은 2개, 라우터 1(R1)은 2개, 링크 #2는 1개, 링크 #3은 1개, 라우터 3(R3)은 2개, 링크 #31은 2개, 링크 #32는 1개의 패킷을 처리한다.

<112> 즉, 도 2와 도 7b를 참조하여, 코어 노드 C0에서 노드 a까지의 패킷전송 경로를 살펴보면, 코어노드 C0에서 링크 #1을 통해 라우터1(R1)를 경유하여 링크 #2를 통해 노드 a까지의 경로가 그려진다.(링크 #1, R1, 링크 #2)

<113> 코어 노드 C0에서 노드 h까지의 패킷 전송 경로를 살펴보면, 코어 노드 C0에서 링크 #1을 통해 라우터1(R1)을 경유하고, 링크 #3을 통해 라우터3(R3)를 경유하고, 링크 #31를 통해 노드 h까지의 경로가 그려진다.(링크 #1, R1, 링크 #3, R3, 링크 #31)

<114> 노드 h에서 패킷을 중계하여 노드 b로 보내주는 경로를 살펴보면, 노드 h에서 링크 #31를 통하여 라우터 3(R3)를 경유하고, 링크 #32를 통해 노드 b까지의 경로가 그려진다.(R3, 링크 #31, 링크 #32)

<115> 따라서, 각 링크 별 라우터별로 패킷이 통과하는 횟수를 계산해보면, 링크 #1은 2회, 라우터1(R1)은 2회, 링크 #2는 1회, 링크 #3은 1회, 라우터3(R3)은 2회, 링크 #31은 1회, 링크 #32는 1회가 되며, 이는 도 7c에서 실선으로 도시되어 있다.

<116> 이에 반하여 종래의 거리 기반 CBT에 의한 결과는 도 7c에 점선으로 대비하여 나타낸 것처럼 각 링크 별 라우터별로 패킷이 통과하는 횟수가 링크 #1은 2회, 라우터1(R1)은 3회, 링크 #2는 2회, 링크 #3은 2회, 라우터3(R3)은 2회, 링크 #31은 1회, 링크 #32는 1회로 대비된다. 종래의 거리 기반 CBT에 의한 결과는 이미 도 3d에서 도시된 바가 있다.

<117> 도 8은 도 2에 도시된 네트워크 토플로지에서 본 발명에 따른 방향성 기반 CBT에서의 멀티 캐스트 패킷의 중계 경로를 나타내는 것으로, 실선은 C0에서 a, h로의 중계 경로를 나타내고, 점선은 h에서 b로의 중계 경로를 나타낸다. 이는 도 4에 도시된 종래의 거리 기반 CBT에서의 멀티 캐스트 패킷의 중계 경로와 대비된다.

<118> 즉, N개의 멀티 캐스트 패킷에 대하여 종래의 거리 메트릭 기반의 CBT에서와 같은 되돌아가는 현상이 발생하지 않아 라우터 R1에서 $3*N$ 개의 패킷이 아닌 $2*N$ 개의 패킷만을 처리한다. 비록 링크 31에서 처리하는 패킷의 수는 증가했으나, 거리 메트릭 기반의 CBT에서의 문제점인 코어 노드가 연결된 라우터 등 특정 라우터나 링크의 패킷 집중현상을 완화시키는 결과를 얻을 수 있다.

<119> 도 9a는 도 2에 도시된 네트워크 토플로지에서 본 발명에 따른 C0, a, b, h, h1의 노드로 구성된 방향성 메트릭 기반의 CBT를 나타내며, 도 9b는 도 9a의 네트워크에서 각 링크 및 라우터에서 처리하는 패킷의 수를 나타낸다. 도 9b를 참조하면, 각각 링크 #1은

2개, 라우터 1(R1)은 2개, 링크 #2는 1개, 링크 #3은 1개, 라우터 3(R3)은 3개, 링크 #31은 3개, 링크 #32는 1개, 링크 #33은 1개의 패킷을 처리한다.

<120> 즉, 도 2와 도 9a를 참조하여, 코어 노드 C0에서 노드 a까지의 패킷전송 경로를 살펴보면, 코어노드 C0에서 링크 #1을 통해 라우터1(R1)를 경유하여 링크 #2를 통해 노드 a까지의 경로가 그려진다.(링크 #1, R1, 링크 #2)

<121> 코어 노드 C0에서 노드 h까지의 패킷 전송 경로를 살펴보면, 코어 노드 C0에서 링크 #1을 통해 라우터1(R1)을 경유하고, 링크 #3을 통해 라우터3(R3)를 경유하고, 링크 #31를 통해 노드 h까지의 경로가 그려진다.(링크 #1, R1, 링크 #3, R3, 링크 #31)

<122> 노드 h에서 패킷을 중계하여 노드 b로 보내주는 경로를 살펴보면, 노드 h에서 링크 #31를 통하여 라우터 3(R3)를 경유하고, 링크 #32를 통해 노드 b까지의 경로가 그려진다.(R3, 링크 #31, 링크 #32)

<123> 노드 h에서 패킷을 중계하여 노드 h1으로 보내주는 경로를 살펴보면, 노드 h에서 링크 #31를 통하여 라우터 3(R3)를 경유하고, 링크 #33을 통해 노드 h1까지의 경로가 그려진다.(R3, 링크 #31, 링크 #33)

<124> 따라서, 각 링크 별 라우터별로 패킷이 통과하는 횟수를 계산해보면, 링크 #1은 2회, 라우터1(R1)은 2회, 링크 #2는 1회, 링크 #3은 1회, 라우터3(R3)은 2회, 링크 #31은 3회, 링크 #32는 1회, 링크 #33은 1회가 되며, 이는 도 9b에 실선으로 표시되어 있다.

<125> 이에 반하여 종래의 거리 기반 CBT에 의한 결과는 도 9b에 점선으로 대비하여 나타낸 바와 같이 각 링크 별 라우터별로 패킷이 통과하는 횟수가 링크 #1은 2회, 라우터

1(R1)은 4회, 링크 #2는 3회, 링크 #3은 3회, 라우터3(R3)은 3회, 링크 #31은 1회, 링크 #32는 1회, 링크 #33은 1회로 대비된다. 이는 이미 도 5b에 도시된 바가 있다.

<126> 이러한 본 발명에 따른 방향성 메트릭 기반 CBT 가입 알고리즘은 다음과 같이 나타낼 수 있다.

<127> Procedure: JoinCBT(h)

<128> $P_i(h) \leftarrow C_0$

<129> $C(P_i(h)) \leftarrow \text{WhoisChild}(P_i(h))$

<130> while $(\text{num}(C(P_i(h)))) \geq \max(C(P_i(h)))$

<131> Find Ny s.t, direction(h, Ny) < direction(h, Nx), Nx, Ny $\in C(P_i(h))$

<132> Find Nx, Nz s.t, direction(Nx, Ny) < direction(Nw, Nx),

<133> $N_v, N_w, N_x, N_y \in C(P_i(h))$

<134> if (direction(h, Ny) > direction(Nx, Nz))

<135> h Join $C(P_i(h))$

<136> c(Nx, Nz) Quit $C(P_i(h))$

<137> c(Nx, Nz) Join $C(p(Nx, Nz))$

<138> return

<139> else

<140> if $(p(h, Ny) == h)$

<141> h Join $C(P_i(h))$

<142> Ny Quit $C(P_i(h))$

```

<143>      Ny Join C(h)

<144>      return

<145>      else

<146>      Increment i, Pi(h) ← Ny

<147>      C(Pi(h)) ← WhoisChild(Pi(h))

<148>      end if

<149>      end if

<150>      end while

<151>      h Join C(Pi(h))

<152>      이 알고리즘을 좀더 상세히 설명하면 다음과 같다.

<153>      Pi(h) ← C0

<154>      C(Pi(h)) ← WhoisChild(Pi(h)) //

<155>      Pi(h) ← C0 : 임의의 노드 h가 CBT에 가입하고자 할 때, 0번째 부모노드를 코아 노드 C0로 한다.

<156>      C(Pi(h)) ← WhoisChild(Pi(h)) : 노드 h는 코아 노드 C0에게 코아 노드 C0의 자식 노드 정보를 요청하여 수신된 자식노드의 정보를 코어 노드 C0의 자식노드로 입력 한다.

<157>      while (num(C(Pi(h))) ≥ max(C(Pi(h)))) : 이 때, 코어 노드 C0의 자식 노드 개수가 최대 수용 가능한 자식 수 이상일 경우에는 다음의 절차를 수행한다.

```

<158> Find Ny s.t., direction (h, Ny) < direction(h, Nx), Nx, Ny $\in C(Pi(h))$: 노드 h와 임의의 노드(노드 Ny, 노드 Nx)간의 방향성을 계산하여 노드 h와 가장 근접한 노드 (Ny)를 찾아낸다.

<159> Find Nx, Nz s.t., direction (Nx, Ny) < direction(Nw, Nx) : 코아 노드의 자식 노드중에서 서로의 방향성이 가장 먼 노드를 찾아낸다.

<160> if (direction(h, Ny) > direction(Nx, Nz)) : 가입하고자 하는 노드 h와 그 노드 h와 가장 근접한 노드 Ny의 방향성이 코어 노드의 자식노드중에서 가장 멀리 떨어져 있는 노드와의 방향성보다 큰지 여부를 판단한다.

<161> h Join C(Pi(h)) : 큰 경우에는 서로 연관성이 없음에 따라 노드 h를 자식노드로 가입시킨다.

<162> c(Nx, Nz) Quit C(Pi(h)) : 코어 노드의 자식 노드인 Nx와 Nz 노드중 코어 노드에서 멀리 떨어진 노드를 코어 노드에서 탈퇴시킨다.

<163> c(Nx, Nz) Join C(p(Nx, Nz)) : 그 탈퇴된 노드를 Nx와 Nz 노드중 코어 노드에서 근접한 노드의 자식 노드로 가입시킨다.

<164> else

<165> if (p(h, Ny)==h) : 노드 h와 노드 Ny중에서 코어노드에 근접한 노드가 노드라면

<166> h Join C(Pi(h)) : 노드 h를 코아 노드의 자식노드로 가입시키고

<167> Ny Quit C(Pi(h)) : 노드 y를 코아 노드의 자식노드로부터 탈퇴시키고

<168> Ny Join C(h) : 노드 y를 코아 노드의 자식노에 자식노드로 가입시킨다.

<169> Increment i, $P_i(h) \leftarrow N_y$

<170> $C(P_i(h)) \leftarrow \text{WhoisChild}(P_i(h))$: 계층을 증가시켜서 다시 노드 y에 대한 가입과정을 진행한다.

<171> 다음으로 CBT를 재구성하는 것을 살펴본다.

<172> 임의의 노드가 CBT에 가입한 후에는 주기적으로 형제 노드와, 부모 노드와 헬로우 패킷을 송수신함으로써 부모 노드와 자식 노드, 그리고 형제 노드들의 상태를 확인 할 수 있다. 일정 시간 헬로우 패킷을 받지 못하면 상대 노드에 장애가 발생한 것으로 간주 한다.

<173> 형제 노드와의 헬로우 패킷을 송수신하는 것을 살펴보면, 최근 수신한 멀티 캐스트 패킷의 송신 번호 정보들을 가진 헬로우 패킷을 형제 노드와 주기적으로 교환, 분석하여 형제 노드와의 방향성 값 계산과 부모 노드와의 통신 이상 여부를 확인한다. 확인한 결과, 부모 노드와의 통신에 일시적인 이상이 발견되면 가장 방향성 값이 작은 형제 노드에게서 패킷을 중계 받을 수 있다. 이와 같이 형제 노드와의 주기적인 통신으로 CBT에 형제 노드간의 메쉬(Mesh) 구조를 만들 수 있다.

<174> 부모 노드와의 헬로우 패킷을 송수신하는 것을 살펴보면, 최근 수신한 멀티 캐스트 패킷의 송신 번호, 형제 노드와의 방향성 정보와 자식 노드가 존재하는 경우 자식 노드와의 방향성 정보등을 주기적으로 부모 노드로 송신한다. 아울러, 형제 노드들의 정보와 삼촌(uncle) 노드들의 정보를 부모 노드로부터 수신한다.

<175> 이때, 부모노드보다 작은 방향성 값을 나타내는 삼촌노드를 발견하면 삼촌 노드에 재가입한 뒤 이전 부모 노드로부터 탈퇴한다.

<176> 한편, 형제 노드와의 헬로우 패킷 교환으로 수신한 멀티캐스트 패킷의 송신 번호를 비교하여 형제 노드보다 작은 송신 번호를 가지고 있으면 부모 노드와의 링크에 장애가 발생한 것으로 판단하고 형제 노드에게 패킷 중계를 요청하여 중계 받는다. 일정 시간 부모 노드로부터 헬로우 패킷을 받지 못하거나 부모 노드와의 링크 장애가 지속되면 할아버지 노드(도면에서 $P_i(h)$, $i=i-1$)에 재가입한다.

<177> 한편, 자식 노드로부터 일정 시간 헬로우 패킷을 받지 못하면 자식 노드에 장애가 발생한 것으로 판단하고 손자 노드들이 재가입할 수 있도록 일정 시간 기다린다. 수용 가능한 자식 노드의 수보다 현재의 자식 수가 적은 경우는 자신노드들로부터 수신한 자식 노드와 손자 노드들과의 방향성을 비교하여 가장 큰 방향성 값을 나타내는 손자 노드를 자식 노드로 재가입시킴으로써 CBT의 계층을 줄여 중계 지연 시간을 줄인다.

<178> 그러면, 본 발명에 따른 방향성 메트릭 기반의 CBT 오버레이 멀티 캐스트와 종래의 거리 메트릭 기반의 CBT 오버레이 멀티 캐스트를 수행하여 코어 노드에서 하나의 멀티 캐스트 패킷을 전체 CBT로 중계하는 과정에서 해당 라우터에 패킷이 집중되는 정보를 비교하여 성능을 평가하도록 한다.

<179> 크게 두가지로 나누어, 전체 CBT의 성능에 영향을 미치는 코어 노드가 연결된 라우터의 처리 패킷수와, 코어 노드외의 가장 많은 패킷을 처리하는 라우터의 처리 패킷수를 산출하여 성능평가를 하도록 한다.

<180> 표 2 내지 표 5는 CBT에 가입하는 실험에서의 가입 노드와 가입순서를 나타내는 표이다.

<181> 【표 2】

순서	1	2	3	4	5	6	7	8	9	10
실1	17	32	29	37	13	3	36	8	1	19
실2	40	38	17	32	37	30	18	13	3	21
실3	16	6	24	10	3	30	29	28	2	1
실4	28	31	15	1	8	33	40	10	20	7
실5	4	29	22	20	35	6	37	15	14	30

<182> 【표 3】

순서	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
실1	25	27	39	35	33	20	1	8	34	7	23	36	26	31	5
실2	12	24	10	16	1	22	31	34	35	9	38	13	25	17	33
실3	10	6	27	38	30	12	14	1	4	33	31	35	29	11	5
실4	22	14	25	33	29	11	13	6	2	16	23	36	30	32	19
실5	4	6	19	11	22	39	5	23	13	38	14	33	21	27	7

<183> 【표 4】

순서	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
실1	7	23	13	2	17	40	25	9	30	16	34	6	14	5	33	32	37	11	39	35
실2	25	6	10	37	14	28	20	12	18	36	27	9	38	8	40	29	35	22	24	11
실3	3	12	18	2	19	7	5	15	28	36	26	37	39	8	35	31	4	23	38	20
실4	34	22	4	17	15	27	6	5	21	35	36	7	23	38	31	37	28	2	9	29
실5	23	27	4	8	20	36	18	17	14	39	38	26	12	5	19	1	29	37	24	33

<184>

【표 5】

순서	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
실1	37	35	2	24	6	15	23	13	39	38	1	19	22	20	32	26	12	11	16	7	10	33	17	40	34
실2	5	26	1	20	3	8	6	35	4	21	7	19	40	25	28	33	34	11	39	17	18	16	30	9	10
실3	21	9	40	17	10	12	4	1	6	34	2	5	29	32	7	38	24	26	15	3	35	39	28	16	13
실4	29	23	12	21	1	9	13	26	5	34	11	20	14	15	38	10	2	39	37	31	19	27	36	28	4
실5	34	11	40	37	7	35	20	30	28	16	25	17	6	9	13	5	15	29	39	38	36	8	2	14	33

<185> 실험과정은 다음의 각 네크워크에서 멀티캐스트 그룹 회원 노드의 수를 증가시키면서 실험을 한다. 그룹 회원 노드는 rand()함수를 이용하여 표 2 내지 표 5와 같이 임의로 선택하고, 또한 임의의 순서로 CBT에 가입한다. 각 회원 노드수에 대하여 5차례의 실험을 반복하고 반복 실험한 오버레이 멀티 캐스트 결과에 대한 평균을 구하여 실험 결과값으로 사용한다.

<186> 실험1. 링 토플로지 네트워크 상에서의 비교

<187> 도 10은 링 토플로지 네트워크를 보여준다. 도 10과 같은 41개의 라우터와 종단 노드들로 구성된 링 토플로지 네트워크에서 표 2 내지 표 5에서의 순서로 10개, 15개, 20개, 25개의 종단 노드들이 CBT에 가입하는 경우를 각각 실험하였다.

<188> 도 11a는 표 2에서 실험1에 대한 거리 메트릭 기반으로 구성된 CBT를 나타내고, 도 11b는 본 발명에 따른 방향성 메트릭 기반으로 구성된 CBT의 예를 보여준다.

<189> 도 11a 및 도 11b를 참조하면 종래의 거리 CBT보다 본 발명의 방향성 CBT의 계층이 하나 작음을 알 수 있다. 이에 따라 패킷을 복사하여 전송하는 중계 과정이 단축되어 그 만큼 빠른 전송이 이루어질 수 있다.

<190> 도 12는 도 11a 및 도 11b에서 종단 노드들과 연결된 각 라우터에서 처리하는 패킷 수를 비교한 그래프이다. 도면에서 점선은 거리 기반 CBT를 나타내고, 실선은 방향성 기반 CBT를 나타낸다. 도 12를 참조하면, 거리 CBT에서는 코어 노드와 노드 1이 연결된 라우터들에 패킷이 집중됨을 알 수 있다. 이에 비해 방향성 CBT는 비교적 완만한 처리 패킷수를 나타낸다. 이는 도 11a의 거리 CBT에서 실제 각 노드들의 위치와 CBT상의 노드 위치가 일치하지 않음으로 인해 노드 1에서 노드 32와 37로 중계된 패킷이 이전의 코어 노드에서 노드 1로 패킷을 전송한 경로를 뒤돌아가기 때문이다.

<191> 도 13a 및 도 13b는 실험 1에서 각 노드들이 CBT에 가입하는 경우를 실험한 결과의 평균처리 패킷수를 나타내는 그래프이다. 도 13a는 코어 노드와 연결된 라우터에서 가입 노드 수별 평균 처리 패킷수를 나타내고, 도 13b는 최대 패킷을 처리하는 라우터의 노드 수별 평균 처리 패킷수를 나타낸다.

<192> 도 13a 및 도 13b의 그래프에서 코어 노드가 연결된 라우터에서 처리하는 패킷의 수가 거리 CBT보다 방향성 CBT에서 약 38% 감소하고, 최대 패킷을 처리하는 라우터의 처리 패킷수도 약 10% 감소하는 것을 알 수 있다. 특히, CBT 전체 멀티캐스트 성능에 영향을 미치는 코어 노드가 연결된 라우터의 처리 패킷 수의 감소로 보다 안정적인 멀티캐스트 통신을 기대할 수 있다.

<193> 실험 2. 선형 토플로지 네트워크 상에서의 비교

<194> 도 14는 선형 토플로지 네트워크를 보여준다. 도 14과 같은 41개의 라우터와 종단 노드들로 구성된 선형 토플로지 네트워크에서 실험 1의 실험과 같은 방법으로 실험하였다.

<195> 도 15a 및 도 15b는 실험 2에서 각 노드별 CBT 가입 실험 결과의 평균처리 패킷수를 나타내는 그래프이다. 도 15a는 코어 노드와 연결된 라우터에서 가입 노드 수별 평균 처리 패킷수를 나타내고, 도 15b는 최대 패킷을 처리하는 라우터의 노드수별 평균 처리 패킷수를 나타낸다.

<196> 도 15a 및 도 15b를 참조하면 실험1과 유사하게 코어 노드가 연결된 라우터에서 처리하는 패킷수가 거리 CBT보다 방향성 CBT에서 평균 약 42% 감소하고, 최대 패킷을 처리하는 라우터의 처리 패킷수도 평균 약 10% 감소하는 결과를 알수 있다.

<197> 실험 3. 성형 토플로지 네트워크상에서의 비교

<198> 도 16은 성형 토플로지 네트워크를 보여준다. 도 16과 같이 41개의 라우터와 종단 노드들로 구성되는 성형 토플로지 네트워크에서 실험1과 같은 방법으로 실험하였다. 도 17a 및 도 17b는 실험 3에서 각 노드별 CBT 가입 실험 결과의 평균처리 패킷수를 나타내는 그래프이다. 도 17a는 코어 노드와 연결된 라우터에서 가입 노드 수별 평균 처리 패킷수를 나타내고, 도 17b는 최대 패킷을 처리하는 라우터의 노드수별 평균 처리 패킷수를 나타낸다.

<199> 도 17a 및 도 17b를 참조하면 코어 노드가 연결된 라우터에서 처리하는 패킷수가 거리 CBT보다 방향성 CBT에서 평균 약 34% 감소하고, 최대 패킷을 처리하는 라우터의 처리

리 패킷수도 평균 약 25% 감소하는 결과와 아울러 CBT의 계층이 줄어드는 효과를 얻을 수 있다.

【발명의 효과】

<200> 이상에서 본 발명에 따른 방향성 메트릭 기반의 CBT를 이용한 오버레이 멀티캐스트를 설명하였다. 이와 같이 패킷의 중계 진행 방향을 고려한 CBT를 생성함으로써, 종래의 거리 메트릭 기반 CBT에서의 특정 라우터에 패킷이 집중되던 문제점을 해결할 수 있게 되었다.

<201> 이러한 효과의 결과로 인하여 특정 라우터에서의 혼잡 발생 가능성을 감소시킴으로써 보다 안정적인 멀티캐스트를 제공할 수 있다.

【특허청구범위】**【청구항 1】**

CBT 기반 오버레이 멀티 캐스트를 위한 CBT 구성방법에 있어서,

CBT에 가입하고자 하는 임의의 종단 노드가 코어 노드에 기가입된 자식 노드 정보를 요청하여 수신하는 단계와,

상기 코어 노드로부터 임의의 자식 노드 정보를 수신한 종단 노드가 자신과 상기 수신된 각 자식 노드와의 방향성을 계산하여 최소의 결과값을 갖는 코어 노드의 자식 노드에 대한 정보를 가입 요구 메시지와 함께 코어 노드에 전송하는 단계와,

상기 코어 노드가 종단 노드로부터 전송된 가입 요구 메시지 및 자식 노드에 대한 정보를 수신하여 해당 자식 노드와 종단 노드의 방향성 계산 결과값을 기가입된 자식 노드간의 방향성 계산 결과값과 비교하여 그 비교 결과에 따라 종단 노드를 해당 자식 노드의 부모 노드 또는 자식 노드로 가입시켜 CBT를 구성하는 단계와,

상기 코어 노드 및 종단 노드가 부모노드, 자식 노드, 형제 노드와 주기적으로 헬로우 패킷을 송수신하여 해당 노드의 상태를 확인하여 그 상태값에 따라 상기 구성된 CBT를 재구성하는 단계를 수행하는 CBT 기반 오버레이 멀티 캐스트를 위한 방향성 기반 CBT 구성방법.

【청구항 2】

제 1항에 있어서, 상기 방향성은,

상기 코어 노드와 이웃하는 임의의 두 노드들간의 거리를 변으로 하는 삼각형에서 이 삼각형의 내외각과 높이를 이용하여 계산되는 CBT 기반 오버레이 멀티 캐스트를 위한 방향성 기반 CBT 구성방법.

【청구항 3】

제 2항에 있어서, 상기 방향성은,
다음의 수식에 의해 계산되는 CBT 기반 오버레이 멀티 캐스트를 위한 방향성 기반 CBT 구성방법.

$$\text{direction} = (\alpha + \beta) * y$$

(단 C_0 : 코아 노드 a_1, a_2 : 임의의 노드라 하고, a_1, C_0, a_2 로 이루어지는 삼각형에서 C_0 과 a_2 가 밑변을 이룰 때, α : 삼각형 a_1, C_0, a_2 에서 변 C_0a_1 와 변 C_0a_2 가 이루는 내각, β : 삼각형 a_1, C_0, a_2 에서 변 C_0a_1 와 변 a_1a_2 가 이루는 외각, y : 삼각형의 높이, direction: 방향성)

【청구항 4】

제 1항에 있어서, 상기 CBT를 구성하는 단계는,
두 노드간의 방향성 값이 작을수록 두 노드는 코어 노드로부터 수신하는 패킷의 진행 방향이 동일하다고 간주하여 부모 노드 또는 자식 노드로 선택하는 CBT 기반 오버레이 멀티 캐스트를 위한 방향성 기반 CBT 구성방법.

【청구항 5】

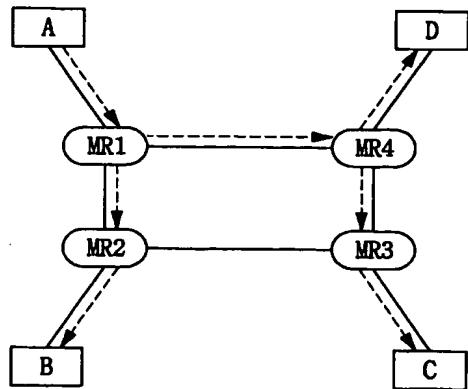
제 1항에 있어서, 상기 CBT를 구성하는 단계는,

상기 코어 노드는, 상기 방향성 계산 결과값을 비교한 결과, 해당 자식 노드와 종단 노드의 방향성 계산 결과값이 기가입된 자식 노드간의 방향성 계산 결과값보다 작은 경우, 종단 노드를 자신의 자식 노드로 가입시키고, 기가입된 자식 노드에 종단 노드의 자식 노드로 재가입하도록 메시지를 전송하는 단계와,

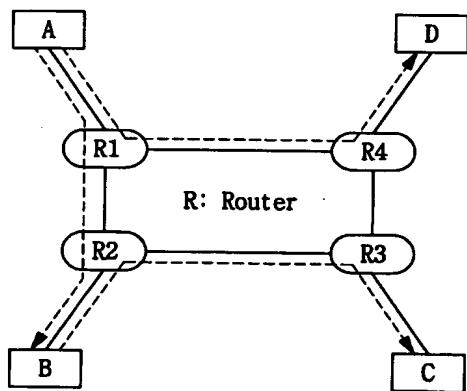
기가입된 자식 노드가 상기 메시지를 수신하여 코어 노드로부터 탈퇴하고 종단 노드의 자식 노드로 가입하는 단계를 수행하는 CBT 기반 오버레이 멀티 캐스트를 위한 방향성 기반 CBT 구성방법.

【도면】

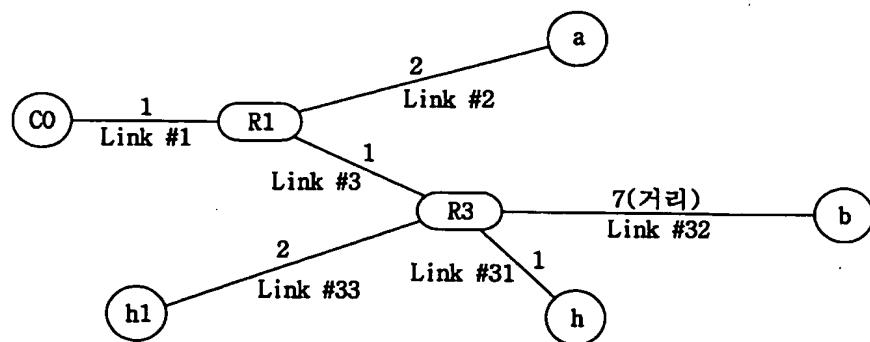
【도 1a】



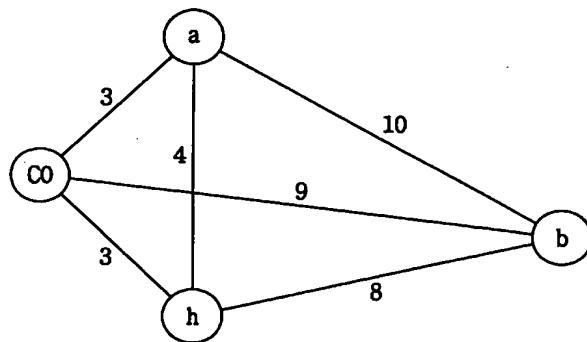
【도 1b】



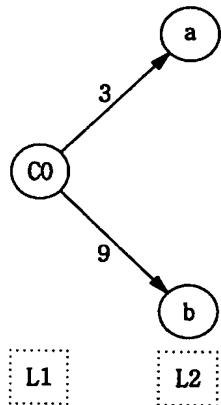
【도 2】



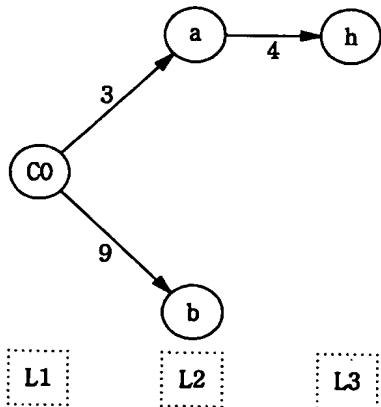
【도 3a】



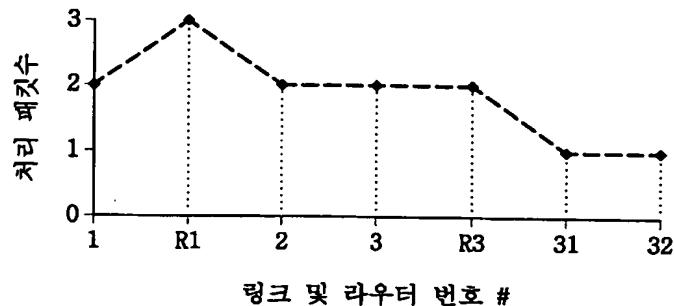
【도 3b】



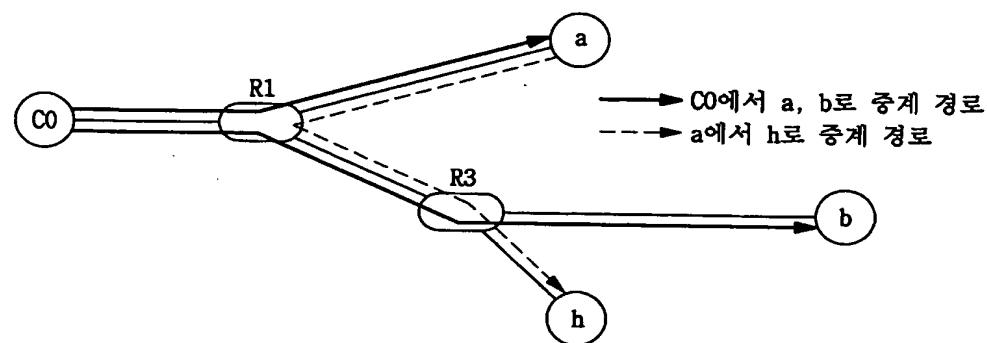
【도 3c】



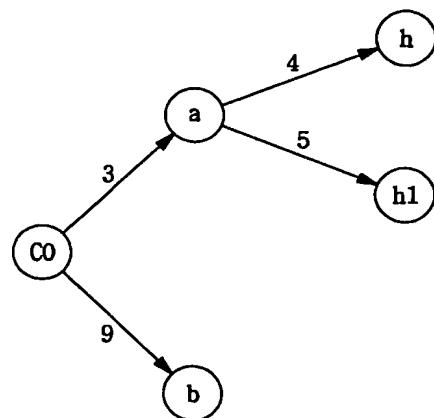
【도 3d】



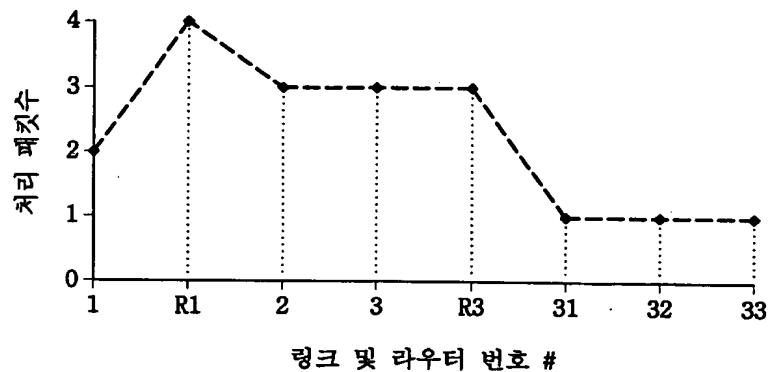
【도 4】



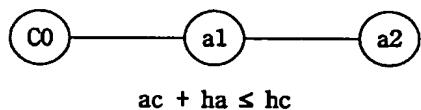
【도 5a】



【도 5b】

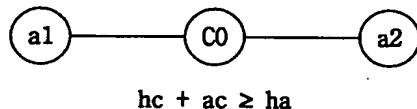


【도 6a】



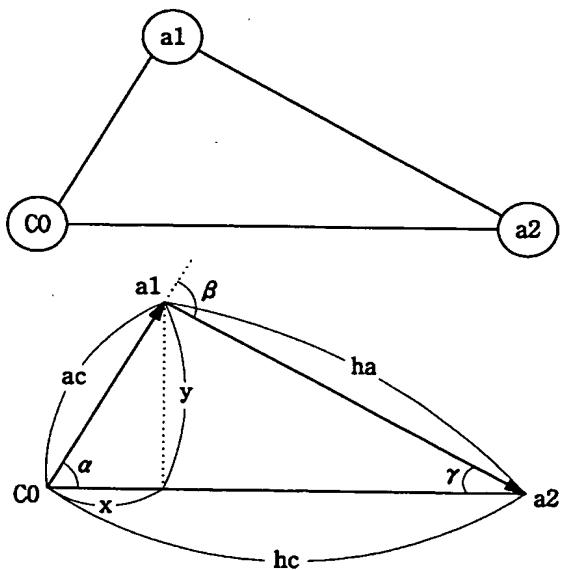
$$ac + ha \leq hc$$

【도 6b】

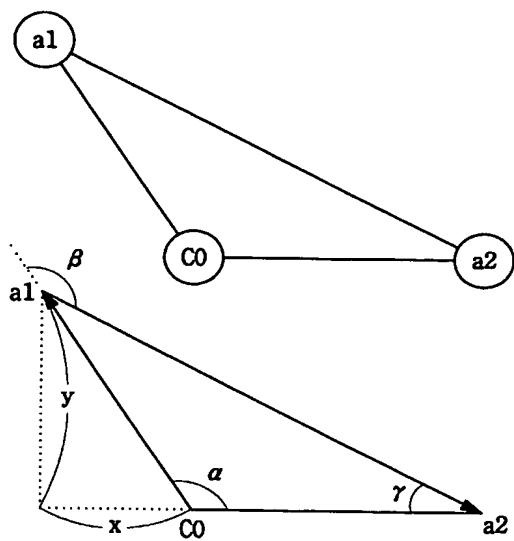


$$hc + ac \geq ha$$

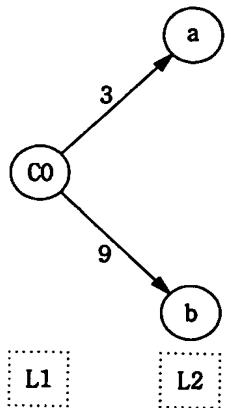
【도 6c】



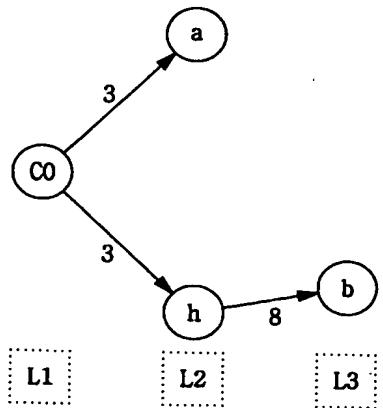
【도 6d】



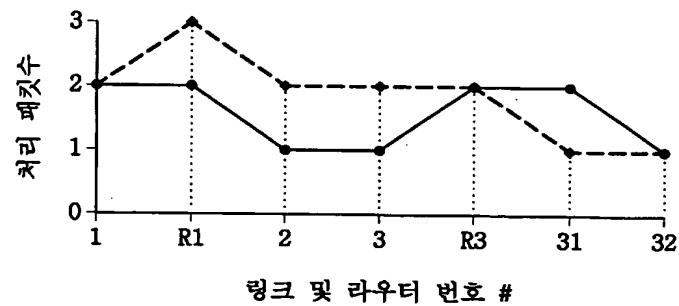
【도 7a】



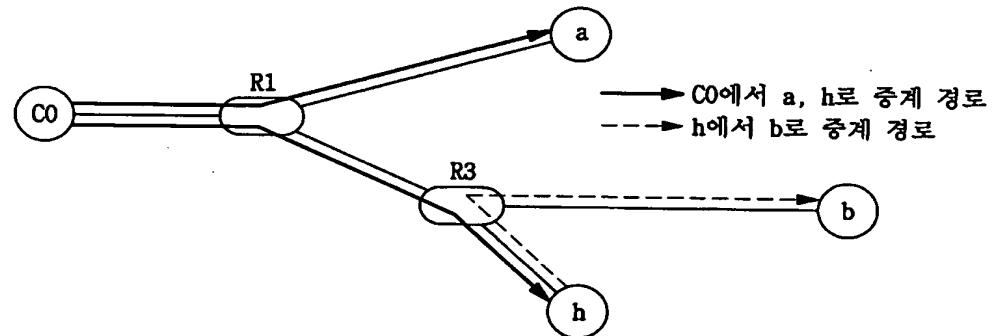
【도 7b】



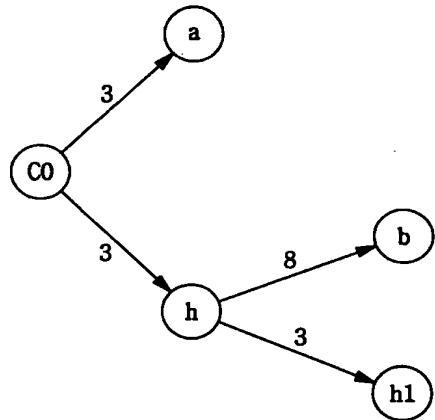
【도 7c】



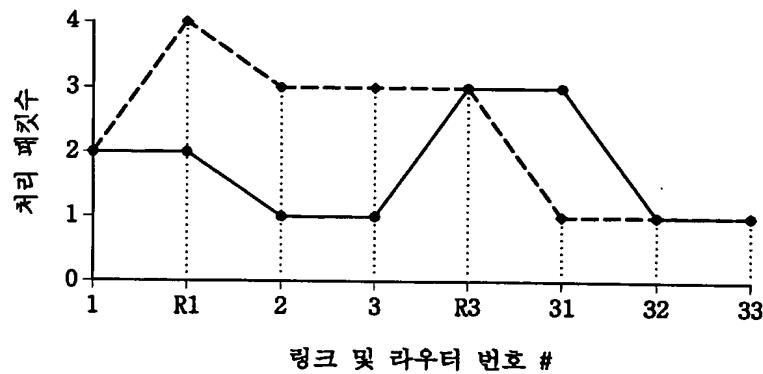
【도 8】



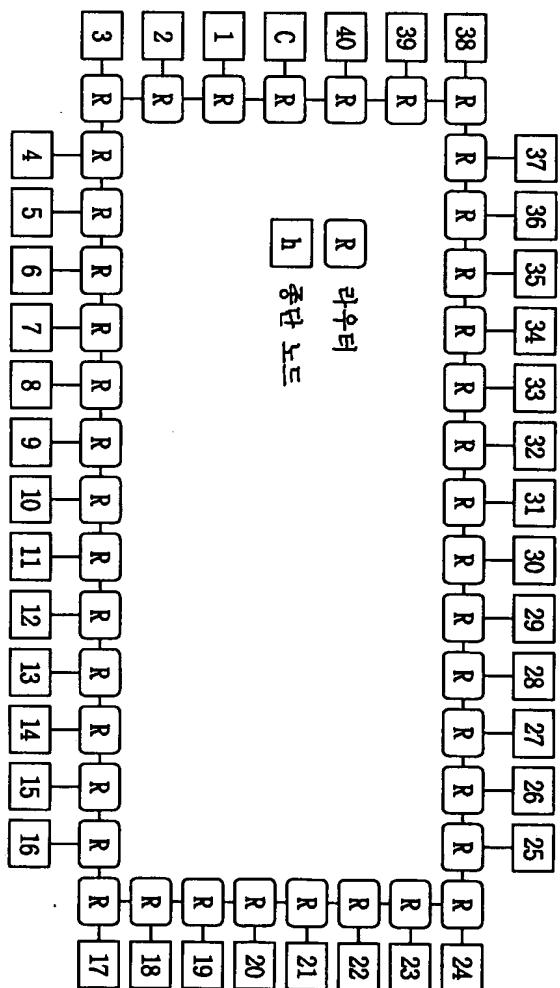
【도 9a】



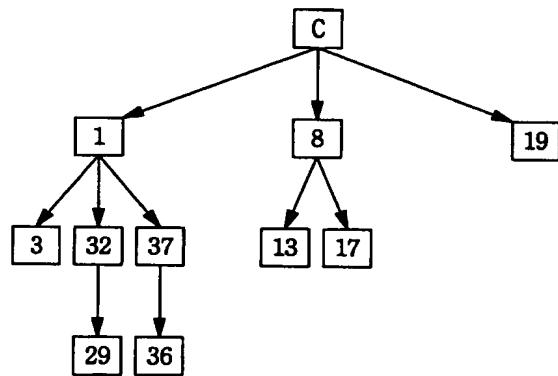
【도 9b】



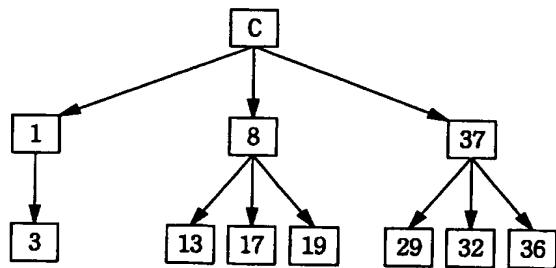
【도 10】



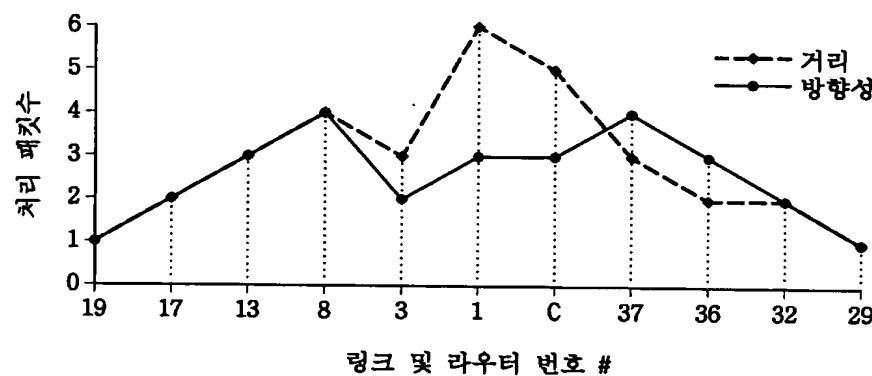
【도 11a】



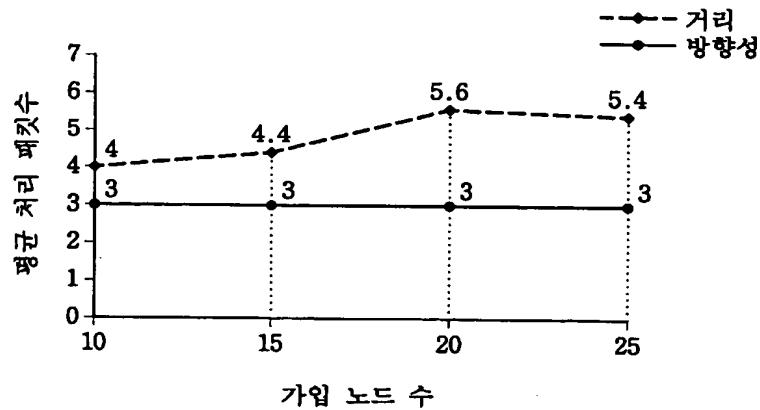
【도 11b】



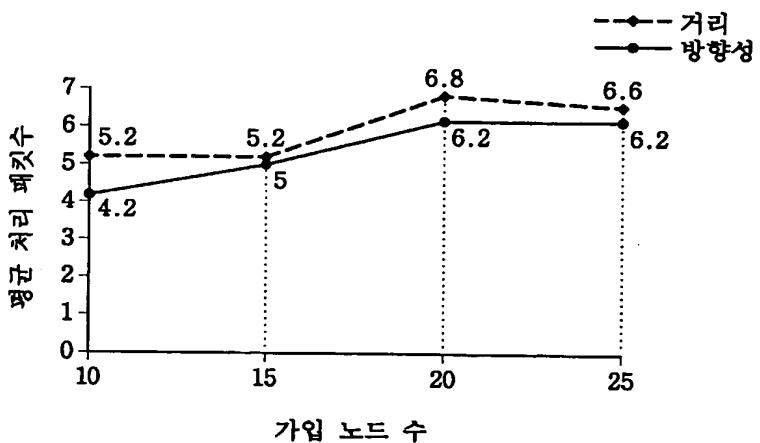
【도 12】



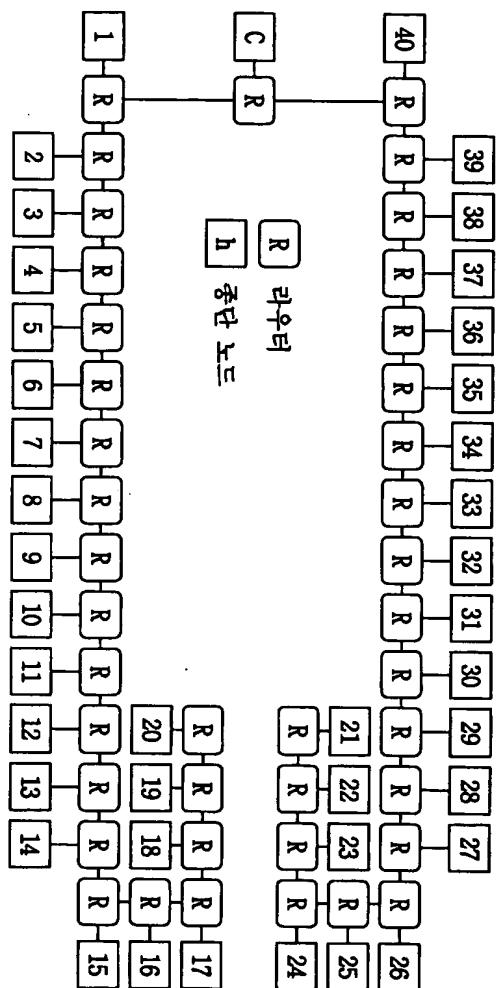
【도 13a】



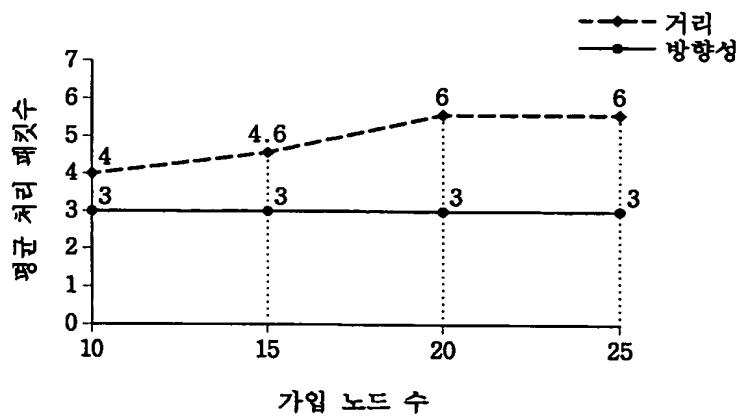
【도 13b】



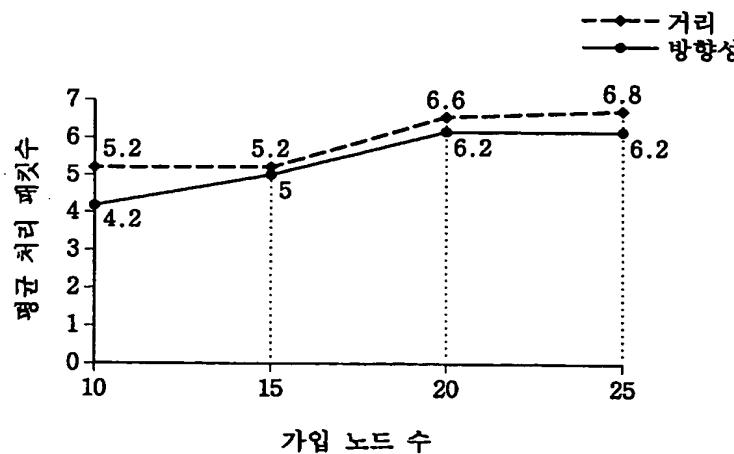
【도 14】



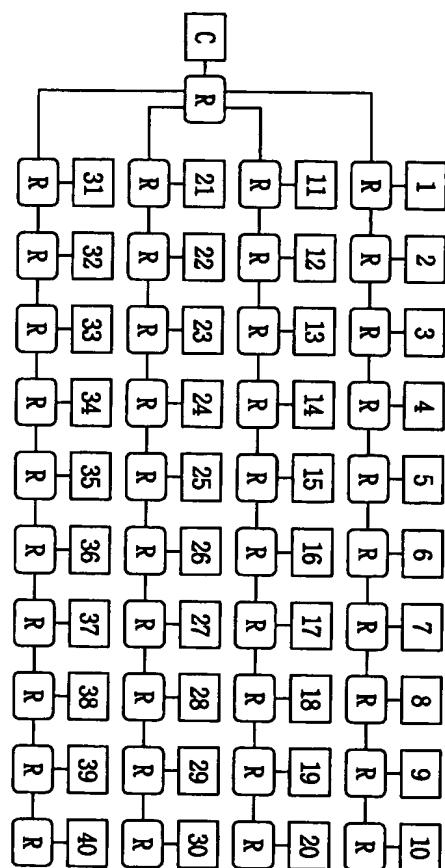
【도 15a】



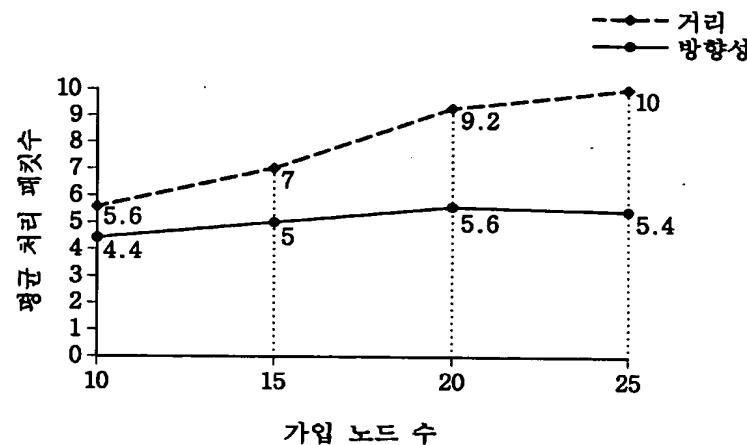
【도 15b】



【도 16】



【도 17a】



【도 17b】

